```
chan up[1:PR](real edge[0:n+1]);
chan down[1:PR](real edge[0:n+1]);
chan diff(real);

process worker[w = 1 to PR] {
  int HEIGHT = n/PR;   # assume PR evenly divides n
  real grid[0:HEIGHT+1,0:n+1], new[0:HEIGHT+1,0:n+1];
  real mydiff = 0.0, otherdiff = 0.0;
  initialize grid and new, including boundaries;
  for [iters = 1 to MAXITERS by 2] {
    # compute new values for my strip
    for [i = 1 to HEIGHT, j = 1 to n]
      new[i,j] = (grid[i-1,j] + grid[i+1,j] +
                  grid[i,j-1] + grid[i,j+1]) * 0.25;
    # send edges of new to neighbors
    if (w > 1)
      send up[w-1](new[1,*]);
    if (w < PR)
      send down[w+1](new[HEIGHT,*]);
    # compute new values for interior of my strip
    for [i = 2 to HEIGHT-1, j = 1 to n]
      grid[i,j] = (new[i-1,j] + new[i+1,j] +
                   new[i,j-1] + new[i,j+1]) * 0.25;
    # receive edges of new from my neighbors
    if (w < PR)
      receive up[w](new[HEIGHT+1,*]);
    if (w > 1)
      receive down[w](new[0,*]);
    # compute new values for edges of my strip
    for [j = 1 to n]
      grid[1,j] = (new[0,j] + new[2,j] +
                   new[1,j-1] + new[1,j+1]) * 0.25;
    for [j = 1 to n]
      grid[HEIGHT,j] = (new[HEIGHT-1,j] +
                        new[HEIGHT+1,j] + new[HEIGHT,j-1] +
                        new[HEIGHT,j+1]) * 0.25;
  }
  compute maximum difference as in Figure 11.4;
}
```

Figure 11.5 Optimized Jacobi iteration using message passing.